Proceedings of the 5th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, Madrid, Spain, February 15-17, 2006 (pp36-42)

# Clarifying Modeling Premises As Explicit Ontologies –Axioms For Conceptual Model Management–

KAZUYA KOYAMA[*1], KAZUHISA SETA[*1], YUSUKE HAYASHI[*2], MITSURU IKEDA[*2]
Department of Mathematics and Information Sciences, Osaka Pref. Univ.
School of Knowledge Science, JAIST
[*1] 1-1, Gakuen-cho, Sakai, Osaka, 599-8531, Japan
[*2] 1-1, Asahidai, Nomi-City, Ishikawa, 923-1292, Japan
http://ks.cias.osakafu-u.ac.jp/, http://www.jaist.ac.jp/ks/labs/ikeda/

*Abstract:* - Modeling is a process of making a model that is an idealized and simplified representation of various objects in the real world and constructed by interpreting and analyzing the target world from a specific viewpoint. Many researchers have paid much attention to building modeling frameworks. The UML is a typical research result acquired by concentrating the know-how on the software development processes. It plays a central role as the foundation that facilitates the model driven system development. But on the other hand, specifying the modeling language is not enough to regulate the fluctuations of modeling. This paper addresses our ontological approach to constructing a framework for accumulating sophisticated modeling premises and illustrates our ontologies as modeling premises ad their operational semantics. In this paper, we firstly describe an overview of our ontology based modeling framework whereby ontology plays a role as modeling premises. Lastly, we illustrate our ontologies and an ontology based conceptual model management function that facilitates the communication among agents.

*Key-Words:* ontology, modeling, ontology based model management, axioms for conceptual model management, modeling premises, identity

## 1 Introduction

Modeling is a process of making a model that is an idealized and simplified representation of various objects in the real world and constructed by interpreting and analyzing the target world from a specific viewpoint. It is notable that a model does not include all the information of the target world, but does include essential information for the development and use of the information system. Therefore, sharing a model provides the following advantages: (i) it contributes to building a deeper consensus among system developers to the design rationales of the system; consequently, (ii) they can build a valid system that satisfies the user's requirements; and (iii) end-users can use the developed system effectively by understanding the design rationales of embedded functions in the system.

We can recognize that the attempt to build a modeling framework such as the UML [1] is an effort to produce an infrastructure that can provide such advantages for model-centered system development.

Nevertheless, it is insufficient to merely develop a modeling language.Modeling language does NOT specify the modeling premises on conceptualization. This will cause (1) fluctuations of modeling and (2) communication gaps among developers through a model. In brief, (1) different models might be built even if the modelers who have same modeling objectives attempt to model identical targets, and (2) the consensus on the model is not built because of the gap separating the intentions of the modeler and the readers' interpretations of the model. Needless to say, all models have their design rationales and a modeling framework must be able to provide them.

We have worked to reduce this problem by taking an approach of clarifying modeling premises and making them explicit as an ontology. Clarifying modeling premises and accumulating them into a machine-readable form contributes to share the meanings of the model among humans and the system. It contributes (1) to reducing the modeling fluctuation because the system can check the validity of the model to the ontologies; it also contributes (2) to ensuring the ability to share design rationales of the model. Model-driven architecture (MDA) tries to provide same advantages and its basic philosophy is common to our approach. The characteristic of our ontological approach in comparison with MDA is that we focus on clarifying the modeling premises at the conceptual level in a machine readable manner whereas MOA does at the system architecture level. Consequently, it engenders the advantage that the system can support model management processes because it can operate the conceptual model based on the ontology [2]. To realize these functionalities, we must clarify the modeling primitives and their operational semantics that play a role as modeling premises. In this paper, we focus on clarifying the operational semantics on knowledge presentation primitives from the viewpoint of conceptual model management based on the analysis of identity.

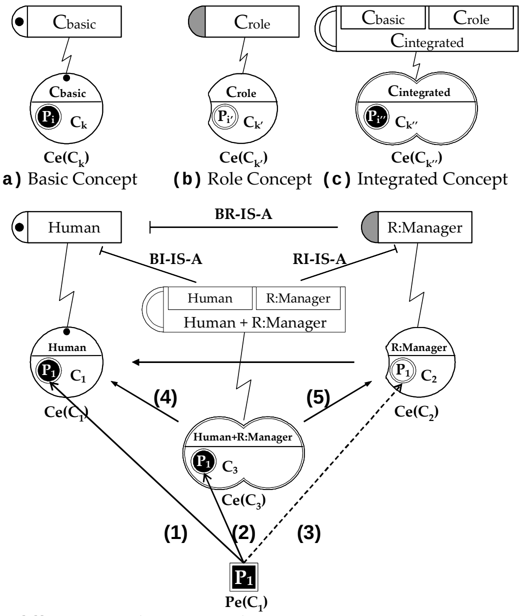
## 2. Modeling Premises

An ontology provides a system of fundamental primitives, concepts and axioms for constructing a model. Therefore, an ontology plays the role of modeling premises: a model is constructed in keeping with the specifications that the ontology defines.

The system can regulate the fluctuations of modeling and operate the model by systemizing the modeling premises explicitly as an ontology and embedding them into the system in a machine-understandable manner, then building a modeling framework whereby one can construct a model according to them.

We have not made the modeling premises for capturing the “identity” explicit; consequently, we have not shared them thus far. This is a fundamental and essential reason for modeling fluctuation. In brief, fluctuation exists in modeling the same target from different modeling objectives or viewpoints. As an example, when we try to model an employee as a subject from the viewpoint of human resource management, we also try to model it from the viewpoint of training. Specifically in this case, we might model the same target as a “manager” from the former viewpoint and as a “learner” from the latter one. This situation presents the problem that we have no consensus as to which objects we should allow to give the identity and how we should represent their oneness. These concepts are collectively referred to as “role concept” modeling [2][3].

Based on an awareness of the issues, we adopt an ontological approach: we attempt to render modeling premises as explicit ontologies on capturing identity .




(d) Conceptual Structures about “Human as a Manager”

**Fig. 1 Model manging process for capturing Identity**

Figure 1 shows our modeling premises for capturing identity. To clarify characteristics of the identity, our research distinguishes two entities: “perceptual entity (Pe)” representing an un-conceptualized entity and “conceptual entity (Ce)” representing a conceptualized one. Furthermore, we respectively denote identities with Pe and Ce as “perceptual identity” and “conceptual identity”.

By introducing these concepts, we clarify the characteristics of the so-called basic concept, role concept, and integration concept for the modeling premises [2].

Integration of the role concept and basic concept can be realized by the inheritance mechanism via “is-a” links. Nevertheless, we require a new aspect of conceptual definitions and is-a links when we examine the use of conceptualization and identity. For instance, consider the relation between “a manager” in the business context and “a human” in the generic context. In this case, we can recognize it not only as a simple relation of “a manager is-a human”, but also as an integration of “a human in a generic context and a role of manager in business task” as multiple inheritances. The difference between these two aspects is clarified by considering the “identity of a manager.” The former can be viewed as different conceptualizations of an identical target perceptual entity at different abstract levels. Therefore, two different conceptual entities must share the same target perceptual entities. For the latter case, on the other hand, we must model three conceptual entities: “a human”, “role of manager”, and “a manager.” Targets of conceptualization and their existence are unified when conceptualizing humans. In contrast, when conceptualizing the “role of manager” and “human as a manager”, the targets of conceptualization and their existence are not unified because the conceptualization target is human.

In our framework, OntoMOF, the conceptual structures mentioned above are modeled as follows. "Human in a generic context" and "human as a manager" simply have a perceptual identity, whereas "role of manager" refers to the perceptual identity of the human that it modifies. Consequently, the "role of manager" is modeled as a role concept, which accompanies "human." "Human as a manager" is modeled as an integrated concept that combines both "human" and "role of manager". We call a concept that has a perceptual identity (human and human as a manager) a basic concept and one that obtains a perceptual identity from a basic concept it modifies (role of manager and human as a manager) an augmentation concept.

Figures 1(a), 1(b) and 1(c) respectively depict the notations of basic concept, role concept and integrated concept. Their upper areas represent concept definitions; their lower ones show their entities. In Fig. 1(a), the small black circle in the left semi-circle represents that the concept is a basic concept. Similarly, in Fig. 1(b) and Fig. 1(c), the gray semi-circle and double line of the semi-circle respectively represent that the concepts are a role concept and an integrated concept.

Figure 1(d) represents conceptual structures of the “human as a manager.” $Pe(P_1)$ simply represents that the perception of existence differs from other existences; $Ce(C_1)$ and $Ce(C_2)$ represent conceptualization of $Pe(P_1)$. In addition, $Ce(C_1)$ is a conceptual perception of $Pe(P_1)$ as a human; $Ce(C_2)$ is a conceptual perception of that human in the role of manager. Furthermore, $Ce(C_3)$ represents conceptual perception as a “human as a manager” that integrates the conceptual perception of $Ce(C_1)$ and $Ce(C_2)$.

The (1), (2) and (3) arrows represent processes of conceptual perception (conceptualization); the (4) and (5) arrows represent relations among conceptual entities. The conceptual entity of a “human as a manager” $Ce(C_3)$ integrates the conceptual information of “human” and “role of manager”. For that reason, we can refer all of their information at $Ce(C_3)$. On the other hand, conceptual perception moves to $Ce(C_1)$ (human) along arrow (4). Therefore, we can not reference the information of “role of manager” $Ce(C_2)$. Similarly, along arrow (5), we can not reference information of humans that are not important to characterize the “role of manager.” The movement of our conceptual perception also changes the available axioms defined at each concept.

Advantages of our modeling premises, described as ontologies, are as follows.

- We can represent our different recognitions on the same target explicitly from different viewpoints.
- In addition, we can elucidate characteristics of the appearance, disappearance and changes: if a basic conceptual entity disappears, the role conceptual entities that accompany it also disappear, whereas a basic conceptual entity does not disappear even if a conceptual role entity that accompanies it disappears. These characteristics are definable in a machine-understandable manner by introducing knowledge representation primitives such as “bi-is-a”, “br-is-a”, “part-of” and so on.

# 3 Ontologies for Model Management

## 3.1 Clarifying Operational Semantics of Knowledge Representation Primitives

The primitives, whose semantics are procedurally implemented, are defined to specify the operational semantics of the models: axioms to specify appearances, disappearances, changes of conceptual entities and propagations of their effects are defined by their combinations. Therefore, operational meanings of the general concepts are also specified. For instance, meanings of the activity concept (tracking time by performing an activity, their effects to the target objects, etc.), relations like is-a and part-of are defined by combinations of the primitives.

Based on analyses of identity and characteristics of each concept, we prepare primitives to set a perceptual identity to a conceptual entity (“occupy”), to integrate or segregate a conceptual basic entity and one for conceptual augmentation (“integrate” and “segregate”, respectively), to extinguish a conceptual entity (“disappear), to manage changes of a conceptual entity (“v-update”), and so on.

These primitives are minimized elements to specify operational semantics of models and play a role in clarifying our modeling premises of how to capture identity and conceptual structures.

Table 1 illustrates axioms of conceptual relations specified based on the primitives, especially on propagations of changes.

**PART-OF:** A PART-OF relation connects a conceptual role entity with a whole concept. It defines the axiom that the conceptual role entity disappears when its whole conceptual entity does so (Table 1(1)). Moreover, it defines that the whole conceptual entity updates its version when its role conceptual entity does so (Table 1(2)).

**BI-IS-A:** A BI-IS-A relation connects a conceptual basic entity with a conceptual integrated entity. It defines the axiom that the perceptual identity of the conceptual integrated entity and the conceptual basic entity should be identical. Moreover, it defines that the conceptual integrated entity disappears when conceptual basic entity does so (Table 1(3)); the conceptual basic entity does NOT disappear when the conceptual integrated entity does so.

**RI-IS-A:** An RI-IS-A relation connects a conceptual role entity with a conceptual integrated entity. It defines the axiom that the conceptual integrated entity disappears when the conceptual role entity does (4) and vice versa.

**Axioms specified by the combination of BR-IS-A and RI-IS-A:** It defines the axiom that the perceptual identities of the conceptual role entity and the conceptual basic entity should be identical. Moreover, by way of the relations of bi-is-a and ri-is-a, the system can realize the model management function as follows: a conceptual role entity (e.g. a manager) disappears when the conceptual basic entity does so (e.g. an employee, a manager, quits the company); a conceptual basic entity (an employee) does NOT disappear even if the conceptual role entity does so (e.g. the employee quits the manager role).

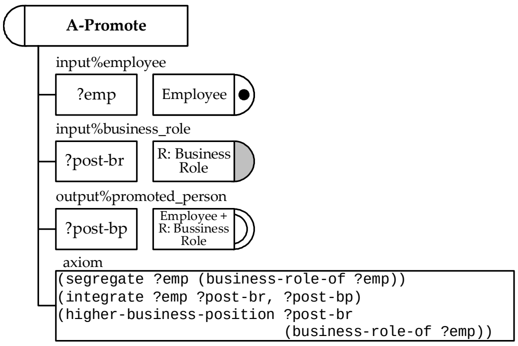
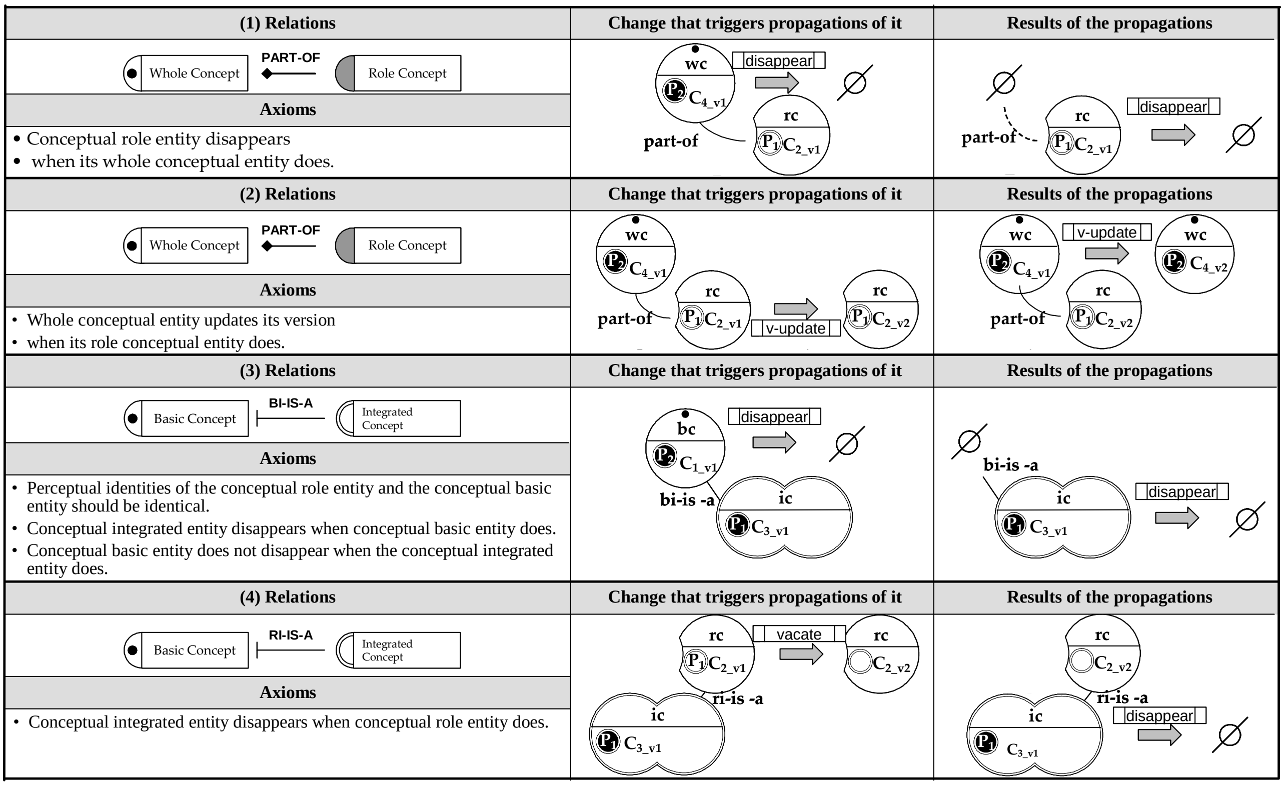



**Fig. 2 Hierarchy of the concepts defined in the core ontology**

By accumulating these kinds of axioms , we can clarify our domain-independent modeling premises in machine readable manner that play a role as specifications of modeling. Moreover, the system can check the validity of the model and manage the



| (1) Relations | Change that triggers propagations of it | Results of the propagations |
|---|---|---|
| Whole Concept —PART-OF— Role Concept | wc, $C_{4\_v1}$; rc, $C_{2\_v1}$; part-of; disappear | part-of; rc, $C_{2\_v1}$; disappear |
| **Axioms** | | |
| • Conceptual role entity disappears • when its whole conceptual entity does. | | |
| **(2) Relations** | **Change that triggers propagations of it** | **Results of the propagations** |
| Whole Concept —PART-OF— Role Concept | wc, $C_{4\_v1}$; rc, $C_{2\_v1}$; part-of; v-update; rc, $C_{2\_v2}$ | wc, $C_{4\_v1}$; v-update; wc, $C_{4\_v2}$; rc, $C_{2\_v1}$; part-of |
| **Axioms** | | |
| • Whole conceptual entity updates its version • when its role conceptual entity does. | | |
| **(3) Relations** | **Change that triggers propagations of it** | **Results of the propagations** |
| Basic Concept —BI-IS-A— Integrated Concept | bc, $C_{1\_v1}$; disappear; bi-is -a; ic, $C_{3\_v1}$ | bi-is -a; ic, $C_{3\_v1}$; disappear |
| **Axioms** | | |
| • Perceptual identities of the conceptual role entity and the conceptual basic entity should be identical. • Conceptual integrated entity disappears when conceptual basic entity does. • Conceptual basic entity does not disappear when the conceptual integrated entity does. | | |
| **(4) Relations** | **Change that triggers propagations of it** | **Results of the propagations** |
| Basic Concept —RI-IS-A— Integrated Concept | rc, $C_{2\_v1}$; vacate; rc, $C_{2\_v2}$; ri-is -a; ic, $C_{3\_v1}$ | rc, $C_{2\_v2}$; ri-is -a; ic, $C_{3\_v1}$; disappear |
| **Axioms** | | |
| • Conceptual integrated entity disappears when conceptual role entity does. | | |

**Table 1:Axiom on conceptual relation defined in Core Ontology**

propagations of changes in it.

### 3.2 Representing Domain Specific Conceptual Meaning Based on the Primitives

The domain specific concepts for modeling the target world can be defined based on the primitives.

Figure 2 shows a conceptual definition of A-Promote that specifies the meaning of promotion activities in the HRM domain.

It defines that a promotion activity inputs a basic conceptual entity of an employee as the target person (input%employee). A target conceptual role entity should be assigned (input%business_role) by performing it, subsequently outputting (output%promoted_person) the integrated conceptual entity of a promoted employee. Furthermore, changes of domain-specific relations by performing the activity are specified as axioms: a promotion activity segregates the employee entity and the pre-business role entity, integrates the employee entity and the post-business role entity. The new business role should be higher position than the pre-business role.

Clarifying the meanings of domain-specific concepts as ontologies contributes to elimination of modeling fluctuations and specifies domain-specific operational semantics of the model in the model layer.

## 4 OntoMOF: Ontology Based Modeling Framework

We can build an ontology based modeling environment that can capture domain-specific meanings by embedding the framework described thus far.

Figure 3 shows a screen image of the modeling environment for the human resource management task in a personnel department.

The window is divided into three views: (A) Organization Map View (OMV), (B) Performance Record View (PRV), and (C) HRM Activity View (HRMV).

OMV is for the MA (Model Author) to model organizational structures of the department by drag-and-dropping (instantiating) the ontologies in Fig. 3(A)(i) on the right side Fig. 3(A)(ii).

In Fig. 3(A), it is specified the company consists of a sales department, application development department and so on, moreover, the application department consists of a manager, chief and assistant

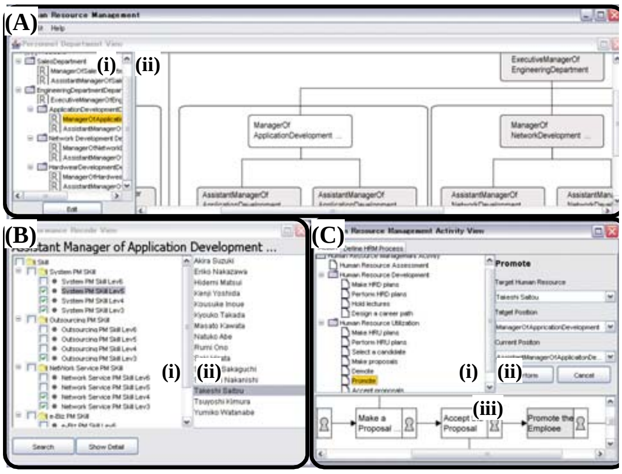



**Fig. 3 Screen image of the modeling environment for the human resource management task in a personnel department.**

manager and so on. Furthermore, it is represented that some employees are assigned to the manager and the assistant manager but no employee is done to the chief. The modeler can refer the predecessors of each role by right mouse clicking on the rectangle of it. This is realized by the model management functions based on the “v-update” primitive.

PRV is for the MA to refer the information of human resources. In the Fig. 3(B), it shows the requirements of skills that the role (e.g. chief) selected in OMV should have. By clicking the “Search” button at the bottom left, the system shows a list of human resources who satisfy the requirements based on the ontology.

The HRMV aids the MA in modeling HRM processes. Fig. 3(C) shows HRM task ontologies that define the activities of human resource assessment, development, utilization, and so on.

Users who perform their daily work according to the model are supported by the system based on the ontologies. For instance, if a person who performs promotion activities clicks on promotion process in Fig. 3(iii), then, based on the concept definition of “A-Promote”. By referring to information on each view, users can perform their activities adequately and the conceptual models are renewed accurately based on ontologies, as shown in Fig. 4. We call this evolutionary conceptual model management.

Furthermore, promotion activity ($Ce(C_{21})$), which is instantiated as the A-Promote concept defined in the ontology, inputs the conceptual basic entity of employee ($Ce(C_{11})$), who plays a role of manager ($Ce(C_{12})$) and the conceptual role entity of the executive manager ($Ce(C_{13})$) at time tin.

Based on the A-Promote definition, the activity segregates the employee ($Ce(C_{11})$) and the manager ($Ce(C_{12})$) and integrates the employee and the executive manager ($Ce(C_{13})$). Moreover, the changes are propagated based on axioms related to conceptual relations.

For instance, the effects of the version change of the executive manager (from $Ce(C_{13\_v1})$ to $Ce(C_{13\_v2})$ are propagated to the whole conceptual entity of the application development department ($Ce(C_{21})$) by way of the part-of relation; it changes its version. Similarly, the effects are propagated to the employee ($Ce(C_{11})$) by way of br-is-a and ri-is-a relations. It changes its version.

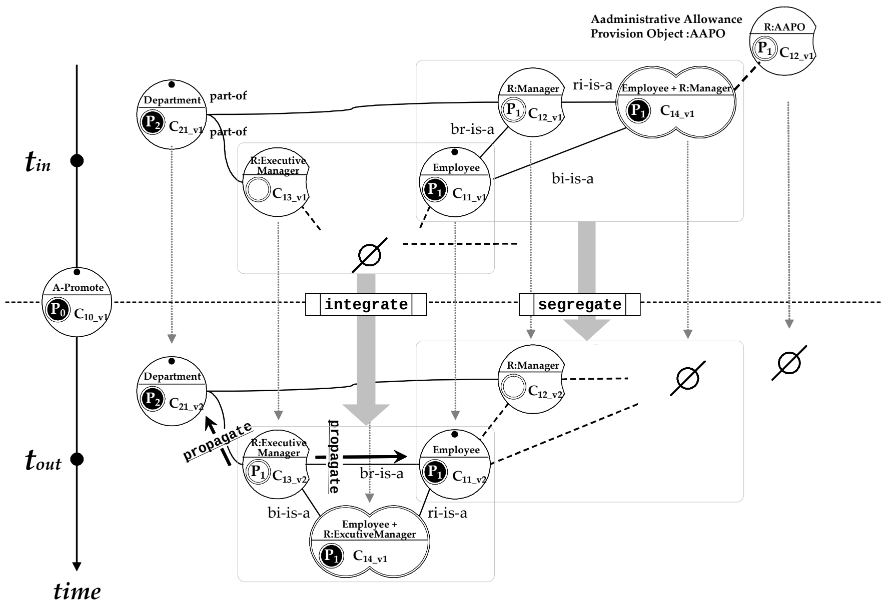



**Fig. 4 Ontology based model management trigged by performing an a-promotion activity**

Propagations that should be addressed in other departments are informed in a message in each environment: “The employee in the application development department is promoted to the executive manager. Thereby, he becomes a person who should be paid a corporate allowance” is shown in the accounting environment. It is noteworthy that the propagation results of the promotion activity in the personnel department are informed appropriately at the conceptual level of the accounting department.

By embedding the ontologies into the basis of the system, one can build valid models according to the ontologies. Furthermore, the system can support encouragement of communications among humans in different departments based on the ontology-based model management function.

Systems with this kind of model management function might be realized using conventional methods. However, in this proposed system, programming codes are connected closely to domain specific concepts; the principle of model management is hidden and becomes implicit.

On the other hand, it is noteworthy that our model management framework, which is realized based on the declarative axioms of core ontology, is quite general. The principle of the functions is represented explicitly.

Thus, the system can be reused for other domains if domain-specific ontologies are built upon the core ontology. Furthermore, the system can explain the mechanism of propagations. This is an important feature that enhances the ability to share models.

## 5 Concluding Remarks

This paper presented an ontology-based modeling framework. We first clarified our underlying philosophy of ontology based modeling by taking our modeling premises on capturing the “identity”. Using that philosophy, the system can regulate the fluctuations of modeling and properly manage the consistency of models. Then, we clarified axioms for conceptual model management by specifying the operational semantics of knowledge representation primitives and illustrated a model management function based on them.

Consequently, an advantage of our approach is that we can advance beyond ad hoc modeling and implementation of a system based on ontology. Our approach provides modeling guidelines and facilitates to establish deeper consensus on the model. Furthermore, we can build a model management function based on the generic level ontology; therefore, it becomes a domain-independent general architecture.

We believe that we can build a foundation of ontology based modeling framework, as a first step, whereby authors can build accurate models without concern for the complicated consistency and evolutionary management and establish deeper consensus on the model based on the ontology.

*References:*

[1] Object Managemant Group (OMG), Unified Modeling Language: Superstructure version2.0 formal/05-07-04, http://www.omg.org/docs/formal/05-07-04.pdf, 2005

[2] Atsushi Matsuo, Kazuhisa Seta, Yusuke Hayashi & Mitsuru Ikeda, An ontological framework for evolutionary model management, Springer, LNCS, No. 3289, pp. 363-376, Shanghai, China, 2004

[3] Jordi Cabot & Ruth Raventós, Roles as Entity Types: A Conceptual Modelling Pattern, Springer, LNCS No. 3288, pp.69-82, 2004